Algorithms for Plane Steiner Tree Problems, DIKU-rapport 98/15

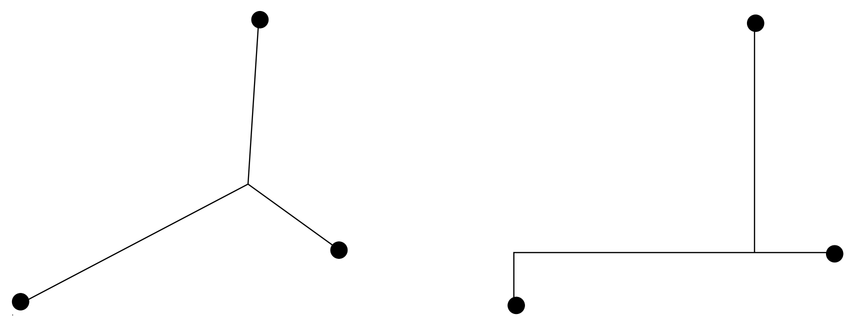
Zachariasen, Martin

Publication date:
1998

Document version
Early version, also known as pre-print

Citation for published version (APA):
Zachariasen, M. (1998). *Algorithms for Plane Steiner Tree Problems, DIKU-rapport 98/15*. Museum Tusculanum.

Algorithms for Plane Steiner Tree Problems



Ph.D. Thesis

Martin Zachariasen

Department of Computer Science (DIKU)
University of Copenhagen
Universitetsparken 1
DK-2100 København Ø, Denmark

March 1998

Abstract

Topological network design is the process of planning the layout of a network subject to constraints on topology. Applications include the design of transportation and communication networks where the construction costs typically are associated with the nodes and/or edges of the network.

The Steiner tree problem is one of the fundamental topological network design problems. The problem is to interconnect (a subset of) the nodes such that there is a path between every pair of nodes while minimizing the total cost of selected edges. Originally, the Steiner tree problem was stated as a purely geometric problem: Given a set of points (terminals) in the plane, construct a tree interconnecting all terminals such that the total length of all line segments is minimized. In the *Euclidean* Steiner tree problem, the length of a line segment is the usual Euclidean (or L_2) distance between the endpoints of the segment. Correspondingly, in the *rectilinear* Steiner tree problem distances are measured using the rectilinear (or L_1) distance metric.

This thesis is about algorithms for solving Euclidean and rectilinear Steiner tree problems. More precisely, new exact and heuristic algorithms for these problems are evaluated by performing extensive computational experiments. The thesis covers the full range of algorithms, in particular for the Euclidean problem: Fast greedy heuristics with an $O(n \log n)$ worst-case running time behaviour, powerful local search algorithms which provide near-optimal solutions quickly, and efficient exact algorithms which solve problem instances with more than 2000 terminals to optimality. The new heuristics provide better solutions faster than any other heuristic proposed in the literature and the new exact algorithm solves problem instances which are more than an order of magnitude larger than those previously solved.

The efficiency of the proposed algorithms stems from a structural property of optimal solutions to these two problems. The optimal Steiner tree breaks into so-called full Steiner trees (FSTs) in which all terminals are leaves. These FSTs are typically very small (seldom span more than six terminals) and have many well-established properties that may be exploited efficiently by using geometric structures.

The thesis is a collection of six research papers. Five of these are on the Euclidean and rectilinear Steiner tree problems, while the last one presents a tabu search algorithm for another geometric problem, the traveling salesman problem in the plane. The thesis begins with a short introduction to algorithms for plane Steiner tree problems.

Preface

This thesis is submitted in partial fulfilment of the requirements for the Ph.D. degree at the Department of Computer Science (DIKU), University of Copenhagen. The thesis has been prepared during the period April 1995 to March 1998 and the work has been supervised by Associate Professor Pawel Winter.

The original research project was entitled "Metaheuristics in Combinatorial Optimization", but since the major effort was spent on studying and developing exact and heuristic algorithms for plane Steiner tree problems, the title and scope of the project was changed accordingly. The thesis consists of a short introduction to the subject and six research papers. The introduction is meant as a moderate level introduction for people with a basis in computer science and/or combinatorial optimization. The introduction is not a primer on Steiner trees but concentrates on *algorithms* for solving plane Steiner tree problems that either already have been implemented or have the potential of being practical.

The six research papers, referred to as [33^A, 37^B, 34^C, 35^D, 30^E, 36^F], are listed among the other references in the introduction (page 15) and separately on page 19. The sixth paper [36^F] falls slightly out of scope with the rest of the thesis since it deals with another optimization problem, the well-known geometric traveling salesman problem. This paper should be considered as an appendix and supports the arguments concerning the performance of local search heuristics. A short Danish summary is given on page 18. The summary is a direct translation of the English abstract.

I am most indebted to my excellent supervisor Pawel Winter, who introduced me to the fascinating world of Steiner trees. Also, I would like to thank David M. Warme for splendid cooperation and for reading drafts of the paper on rectilinear full Steiner trees. I would like to thank the members of the Algorithmics and Optimization Group at DIKU for moral support and valuable discussions. Similarly, my colleagues in Eindhoven — with whom I stayed for eight months — are warmly thanked for their friendliness and for letting me participate in professional and social activities. I thank Dávur Sørensen for commenting on my English in the introduction. Finally, but not least, I would like to thank my wife Elin for her support and understanding.

Copenhagen, March 1998

Martin Zachariasen

Contents

1 Introduction

Transportation networks play an increasingly important role in today's society. Networks for transporting people, goods and bits form the backbone of almost all business and trade activities. The design of such networks is therefore an important political and engineering task.

The costs associated with a network are the *construction* costs which often can be assumed to be proportional to the total length of the network and the *flow* costs which depend on the length and the amount of flow through the network. The design of a network involves finding a balance between these two cost factors, since a network with a small total length generally does not result in a most cost-effective network for the flow problem. However, a network which minimizes construction costs will often form a natural basis for designing the final network which takes flow costs into account (by, e.g., adding direct links between locations having a large inter-flow).

A simplified mathematical model of the construction problem is the following: Assume that the locations correspond to points in the plane and that the length of a link is the Euclidean distance between the endpoints of the link. The problem is now to interconnect the given points, also denoted *terminals*, such that the total length of all line segments is minimized.

It has been known since antiquity that the shortest interconnection of two points is a straight line segment (Figure 1a). The problem of interconnecting three points was proposed by Fermat and solved independently by Torricelli and Cavalieri in the 17th century. Depending on the configuration of the three points, the optimal solution may contain a junction, a so-called *Steiner* point (Figure 1b). Thus it is not sufficient to consider direct connections between terminals only.

The general problem involving an arbitrary number of terminals has been attributed to the 19th century mathematician Jakob Steiner, although he was mainly concerned with the problem of seeking a single point whose connections to the terminals had the shortest possible total length. The general problem is denoted the *Euclidean Steiner tree problem* and optimal solutions, which must be trees, *Steiner minimum trees (SMTs)*, as shown in Figure 1c.

Since Steiner points apparently may be placed anywhere in the plane, it is not obvious that a (finite-time) algorithm exists for the problem. However, using fundamental structural properties of SMTs (described in Section 2), Melzak [20] gave the first algorithm for solving the problem[1].

The Steiner tree problem has been generalized to other metrics and higher di-

[1] Assuming that algebraic operations, including taking square-roots, can be performed in constant time.

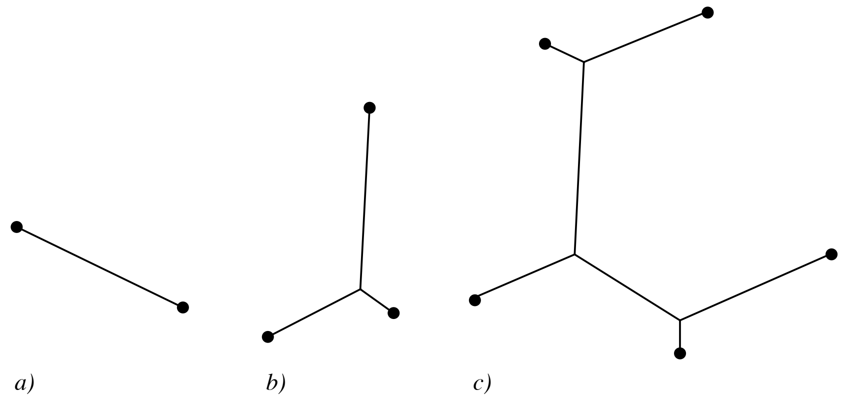


Figure 1: Euclidean Steiner minimum trees.

mensions. One particular case has received considerable attention because of its applications in the design of very large-scale integrated (VLSI) circuits. The design of VLSI chips has a sequence of stages in which a high-level system description is transformed into a set of mask geometries for fabrication. The final stage, the physical design, has a *placement* step in which functional units are placed on the surface of the chip and a *routing* step which interconnects specified terminals of the functional units. The total length of wires used for interconnecting the terminals should be minimized subject to the condition that the wires are embedded in vertical and horizontal directions only. This is equivalent to solving the *rectilinear Steiner tree problem* in which distances between points are measured using the rectilinear (or Manhattan) distance metric, which is the sum of differences in x- and y-coordinates (Figure 2).

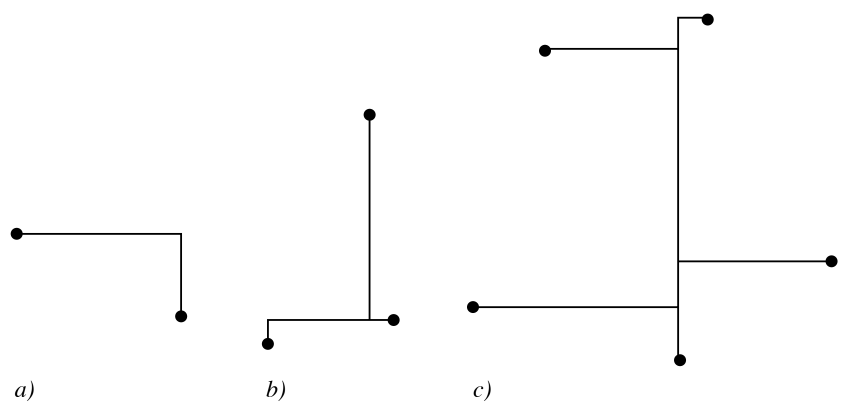


Figure 2: Rectilinear Steiner minimum trees.

Hanan [12] presented the first thorough study of the rectilinear problem; one of his contributions was to show that the Steiner point candidates may be confined to the vertices of what is now known as the Hanan grid graph (see Section 2). The number of vertices in this graph is (only) quadratic in the number of terminals and this is in sharp contrast to the superexponential number of Steiner point candidates which essentially result when applying the Melzak-algorithm to the Euclidean problem.

The Euclidean and rectilinear Steiner tree problems can be regarded as being special cases[2] of the *Steiner tree problem in graphs*: Given an undirected graph $G = (V, E)$ with positive edge-weights and a non-empty set $Z \subseteq V$ of terminals, the problem is to find a shortest tree interconnecting Z. Note that the length of the tree is the sum of weights of the edges in the tree; furthermore, these edge-weights do not need to be related to any familiar distance metric.

All Steiner tree problem variants mentioned here have been shown to be NP-hard [8, 9] and therefore no polynomial time algorithm is believed to exist. This thesis is about practical algorithms for solving the Euclidean and rectilinear Steiner tree problems in the plane. Exact algorithms are presented in Section 3 and heuristics, which in general only find suboptimal solutions, are covered in Section 4. The main emphasis is put on algorithms that have been implemented and empirically evaluated, with particular attention to the new algorithms given in this thesis. In order to widen the perspective, algorithms that recently have been proposed (but not have been implemented yet) and appear to be promising, are also discussed.

2 Structural Properties of SMTs

Many structural properties are known for plane SMTs [10, 15]. A list of the most essential properties are given here; these properties are applied in the construction of both exact and heuristic algorithms.

Assume that we are given a finite set Z of n terminals (points in the plane) and that we would like to construct an Euclidean SMT (ESMT) or a rectilinear SMT (RSMT) for Z. For any pair of points $u = (u_x, u_y)$ and $v = (v_x, v_y)$ in the plane, the distance in the L_p-metric, $1 \leq p \leq \infty$, between u and v (or simply the L_p-distance) is $\|uv\|_p = (|u_x - v_x|^p + |u_y - v_y|^p)^{1/p}$. As special cases we have the Euclidean L_2-distance

$$\|uv\|_2 = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}$$

and the rectilinear (or Manhattan) L_1-distance

$$\|uv\|_1 = |u_x - v_x| + |u_y - v_y|.$$

[2]Since the number of Steiner point candidates is finite for both problems.

Angle and Degree Conditions

All edges in an ESMT must meet at angles that are 120° or greater; this is a straightforward consequence of the optimality of the tree and one of the first general properties identified for Steiner trees. One implication is that Steiner points are incident with exactly three edges making 120° with each other (Figure 3a). More generally, Steiner points in SMTs under the L_p-metric, $1 < p < \infty$, have degree three and edges making more than 90° with each other [18].

It is always possible to find an RSMT with edges belonging to the Hanan grid [12]. The Hanan grid is obtained by drawing horizontal and vertical lines through all terminals (Figure 3b). Therefore, RSMTs can be embedded in the plane by drawing horizontal and vertical line segments such that edges meeting at vertices make either 90° or 180° with each other. A Steiner point therefore has either degree three or four. Since Steiner points in SMTs under *any* metric have degree three or more an SMTs for n terminals has at most $n-2$ Steiner points.

Full Steiner Topologies and Full Steiner Trees

A topology is a description (graph) of the connections (edges) between terminals and Steiner points (vertices). A Steiner topology is a topology for which all Steiner points have degree three. A Steiner topology is *full* if the number of Steiner points is maximal ($n-2$); note that in a full Steiner topology all terminals are leaves.

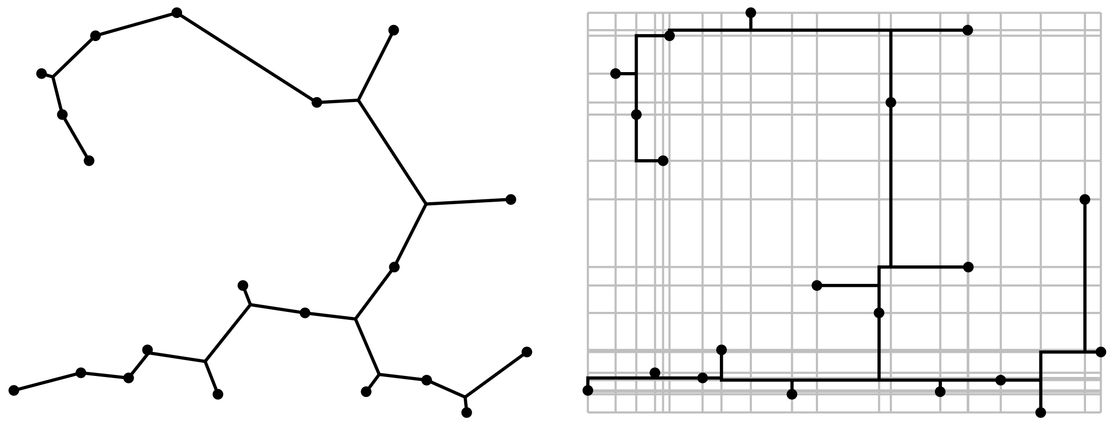
The number $f(n)$ of full Steiner topologies (each defining a topologically different tree) with n terminals and $n-2$ Steiner points is given by

$$f(n) = \frac{(2n-4)!}{2^{n-2}(n-2)!}$$

The function f is superexponential in n, that is, it increases faster than an exponential function (e.g., $f(4) = 3$, $f(6) = 105$ and $f(8) = 10395$).

The shortest tree with a given topology is called the *relatively* minimal tree. This relatively minimal tree can be found efficiently since the minimization problem is convex in the locations of the Steiner points (a local minimum is also a global minimum). Note that the relatively minimal tree may have zero-length edges, i.e., Steiner points overlapping with terminals or with each other.

An SMT is a relatively minimal tree for some full Steiner topology for Z. Zero-length edges in the relatively minimal tree correspond to breaking the full Steiner topology for Z into smaller full topologies. The relatively minimal trees for each of these smaller full topologies are denoted *full Steiner trees (FSTs).* We say that an SMT is a union of FSTs in which every leaf is a terminal and all other nodes are Steiner points (Figure 3).



a) *b)*

Figure 3: Euclidean and rectilinear Steiner minimum trees for a set of terminals. The optimal solutions break into small full Steiner trees (FSTs). The rectilinear tree may be confined to the Hanan grid.

Minimum Spanning Trees and the Steiner Ratio

A minimum spanning tree (MST) for the terminals in Z is a shortest network which spans Z and does not introduce Steiner points. Euclidean MSTs (EMSTs) and rectilinear MSTs (RMSTs) can be constructed in $O(n \log n)$ time [21]. The infimum (over all terminal sets Z) of the ratio between the SMT length and MST length is denoted the *Steiner ratio*. Gilbert and Pollak [10] conjectured in 1968 that this value was $\sqrt{3}/2$ for the Euclidean problem; a proof was given by Du and Hwang in 1992 [7]. The Steiner ratio for the rectilinear problem is 2/3 [14]. Therefore, an ESMT (resp. RSMT) is at most 13.4% (resp. 33.3%) shorter than an EMST (resp. RMST).

3 Exact Algorithms

Historically, exact algorithms for the Euclidean and rectilinear problems have developed quite independently and in different directions. One of the reasons is the straightforward reduction of the rectilinear problem to the Steiner tree problem in graphs via the Hanan grid graph. This allowed existing exact algorithms for the general graph problem to be applied; these algorithms used, e.g., mathematical programming formulations which were solved by using standard techniques (Section 3.1). Thus, these algorithms did not exploit the geometry of the problem.

For the Euclidean problem two approaches have essentially been suggested. The first one is based on generating and concatenating full Steiner trees (Section 3.2) and the second on enumerating full Steiner topologies for Z (Section 3.3). The former approach is currently by far the most efficient practical algorithm for computing Euclidean *and* rectilinear Steiner minimum trees.

3.1 Reduction to the Steiner Tree Problem in Graphs

The rectilinear Steiner tree problem can be solved as a graph problem with at most n^2 vertices and $2n(n-1)$ edges (for a comprehensive survey of exact algorithms for the Steiner tree problem in graphs, see [15]). Early algorithms for the general graph problem include the spanning tree enumeration algorithm of Hakimi [11] and the dynamic programming algorithm of Dreyfus and Wagner [6]. More recent algorithms use integer programming formulations which are solved by branch-and-cut.

The best exact algorithms [19, 17] can solve problem instances with several thousand vertices and edges, but on rectilinear instances (i.e., the Hanan grid graph) the linear programming relaxations become highly degenerated; therefore, the current limit for these algorithms is approximately 40 terminals. However, specialized graph reduction methods can be applied to the Hanan grid graph before exact algorithms for the graph problem are applied [32, 35^D], and this has the potential of increasing the solvable range considerably.

3.2 Generating and Concatenating Full Steiner Trees

The first exact algorithm for the Euclidean problem, the Melzak-algorithm, used the following basic framework. Subsets of terminals are considered one by one. For each subset, all its full Steiner trees (FSTs) are determined one by one, and the shortest is retained. FSTs are then concatenated in all possible ways to obtain trees spanning Z, the shortest being an SMT.

Melzak's main contribution was to describe an algorithm that, given a subset of terminals and a full Steiner topology for the subset, finds an FST which has the specified topology (provided that such an FST exists). Since the number of (subset, topology) pairs is superexponential, this basic framework has limited practical value.

Winter [31] suggested a departure from this framework. He observed that substantial improvements can be obtained if FSTs are generated across various subsets of terminals. The idea is to test whether a given *sub*topology can be extended to a topology which may appear in an SMT. For instance, consider two terminals

z_i and z_j sharing a Steiner point s. This is a partial specification of a larger topology. In fact, there exists a superexponential number of topologies having this subtopology.

Several tests can be applied in order to identify and prune away subtopologies that cannot appear in any SMT, e.g., tests based on the so-called *lune property*. A lune of a line segment uv is the intersection of two circles both with radius $\|uv\|_p$ and centred at u and v, respectively (Figure 4). A necessary condition for the line segment uv to be in any SMT is that its lune contains no terminals.

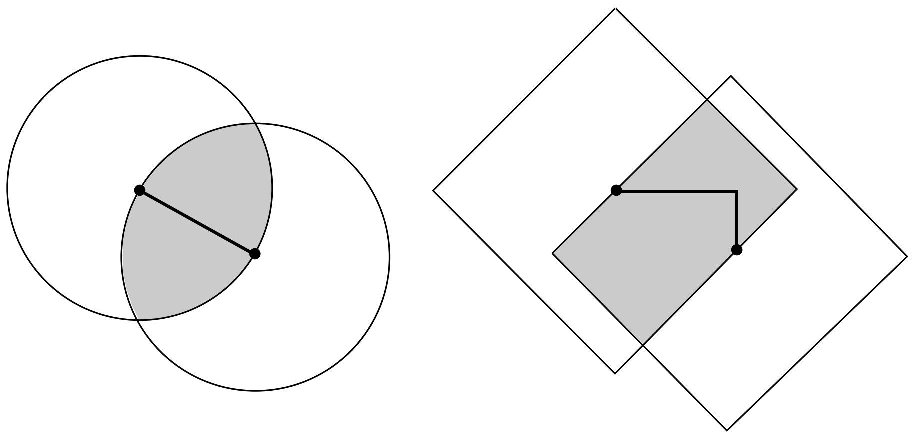


Figure 4: Euclidean and rectilinear lunes.

By using the lune property it is possible to test if a subtopology in which z_i and z_j share a Steiner point s can appear in any SMT (Figure 5): If s cannot be placed anywhere without violating the lune property, all topologies having this subtopology can be pruned away.

Sophisticated pruning techniques were described by Winter and Zachariasen [33^A]. A similar strategy has been applied to the rectilinear problem [35^D]. Rectilinear FSTs are known to have a very simple topology which allows very efficient generation. Furthermore, the lune property can be extended to more complex empty regions.

The number of FSTs surviving the generation phase is typically only *linear* in the number of terminals [33^A, 35^D, 30^E]. Given the set $\mathcal{F}$ of FSTs, the *concatenation* problem is to identify a subset $\mathcal{F}^* \subseteq \mathcal{F}$ such that the FSTs in $\mathcal{F}^*$ interconnect Z and have minimum total length. Early algorithms for solving the concatenation problem used simple backtrack search or dynamic programming. Powerful methods for reducing the set of FSTs before entering the concatenation phase were given in [33^A]. These methods more than doubled the size of instances that could be solved to optimality.

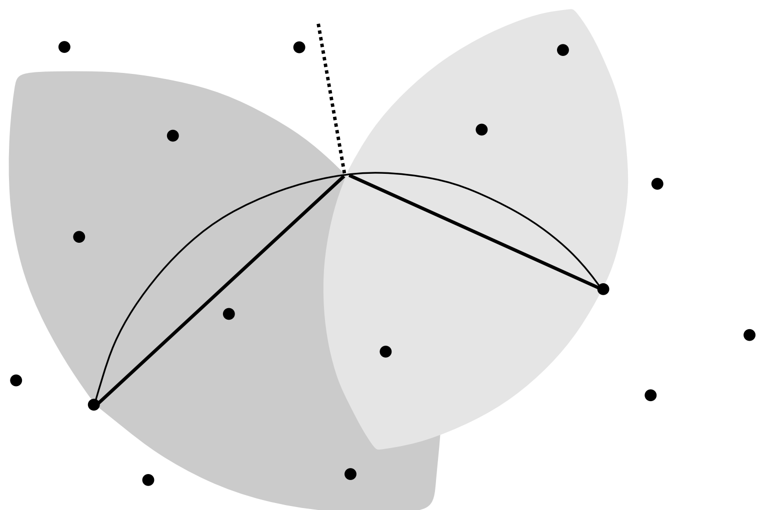


Figure 5: Pruning subtopologies. The Steiner point is restricted to the *Steiner arc* since the incident edges are required to make 120° with each other. The lune property is violated for all possible Steiner point locations.

Recent results of Warme [29] have improved the concatenation dramatically. He noticed that the concatenation of FSTs can be formulated as a problem of finding a minimum spanning tree in a hypergraph with terminals as vertices and subsets spanned by FSTs as (hyper)edges. He solved this problem using branch-and-cut. Euclidean and rectilinear problem instances with as many as 2000 terminals can today be solved to optimality by using the before mentioned FST generation algorithms [33^A, 35^D] in conjunction with Warme's concatenation algorithm [30^E].

3.3 Enumerating Full Steiner Topologies

Recall that an SMT is a relatively minimal tree for some full Steiner topology. One approach to finding an SMT is to enumerate all full Steiner topologies and to compute a relatively minimal tree for each of these; a shortest relatively minimal tree is then an SMT. This approach avoids the concatenation of FSTs completely. In the following some approaches along this line, all related to the Euclidean problem, are discussed.

The Melzak-algorithm constructs a relatively minimal tree for a given full Steiner topology — but only if it does not degenerate (corresponding to a tree with zero-length edges). The *negative edge* algorithm [28] extends the Melzak-algorithm, enabling it to find relatively minimal trees for degenerate topologies of a given full topology as well. Unfortunately, the running time of this algorithm has not been proven to be polynomial.

Hwang and Weng [16] constructed an $O(n^2)$ time algorithm for the problem of finding a relatively minimal tree for given full Steiner topology. The *luminary* algorithm has two stages as the Melzak-algorithm, but it involves keeping track of much more information during the construction. The algorithm is quite complicated and has as yet never been implemented. Therefore, it is still an interesting open question whether this algorithm is practical.

A much simpler numerical algorithm, which also can be generalized to higher dimensions, was given by Smith [26]. It solves the associated convex optimization problem by using an iterative numerical algorithm. The locations of the Steiner points are iteratively updated using a linear-time update procedure which moves all Steiner points simultaneously. The procedure converges for all initial choices of Steiner point locations (except for a point set of measure zero). An implementation of the algorithm verified its effectiveness; it was conjectured that the algorithm needs $O(n/\epsilon)$ iterations to reach a length within ϵ of optimal. Thus, in order to get below a fixed machine precision, a running time of $O(n^2)$ is supposedly required.

As noted in Section 2, the number of full Steiner topologies is superexponential in the number of terminals. Therefore, the algorithms presented in this section cannot be practical unless a substantial number of full Steiner topologies can be pruned away simultaneously. Smith noted that in the planar case "only" an exponential number of full Steiner topologies need to be considered since all other topologies result in self-intersecting relatively minimal trees. Smith also gave simple tests to discard a set of topologies based on computing a lower bound on all relatively minimal trees for the set of topologies. However, these tests had a limited effect on the number of topologies considered. Algorithms based on enumerating full Steiner topologies are currently able to solve instances with up to 15-20 terminals within a reasonable amount of time.

4 Heuristics

The hardness of the Euclidean and rectilinear Steiner tree problems has resulted in an abundance of heuristics for these two problems. Contributions go back to the early 1970s with an increasing interest in the rectilinear problem because of its important applications.

In theoretical computer science there is a more or less well-defined distinction between heuristics and *approximation* algorithms. An approximation algorithm is essentially an heuristic which has a well-established performance guarantee. More precisely, we say that a polynomial time algorithm that guarantees a solution which is at most[3] a factor r from optimum is denoted an r-approximation algorithm (where $r > 1$ is a constant); for more details on approximation algorithms the reader is referred to [13].

All heuristics discussed in the following are actually approximation algorithms since they provide solutions that are no worse than a minimum spanning tree (MST) for Z. A consequence of the Steiner ratio theorem is that an EMST (resp. RMST) is at most a factor $2/\sqrt{3}$ (resp. 3/2) longer than an ESMT (resp. RSMT). In fact, most heuristics are based on improving the MST by adding Steiner points in a greedy manner (Section 4.1). More sophisticated heuristics use local search to obtain high-quality solutions (Section 4.2).

On the theoretical front, there has recently been a major development in the construction of so-called polynomial time approximations schemes (PTAS) for the Steiner tree problem and other geometric problems. For every fixed $\epsilon > 0$ a PTAS provides a $(1 + \epsilon)$-approximation algorithm — intuitively this means that for any fixed percentage, say 1%, there exists a polynomial time algorithm that returns solutions that at are most 1% from optimum. These interesting contributions are highlighted in Section 4.3.

4.1 Greedy Heuristics

The problem of computing an SMT can be reformulated as a problem of finding a set of Steiner points S such that the minimum spanning tree for $Z \cup S$ is as short as possible (over all possible sets S). Therefore, a natural greedy algorithm is to start with an empty set S and to add Steiner points to S in a greedy manner.

For the rectilinear problem, Steiner points may be confined to vertices of the Hanan grid giving at most n^2 Steiner point candidates in each iteration. For the Euclidean problem there are no easily identified Steiner point candidates. Thompson [27] proposed to look for edges in the MST for Z meeting at angles less than 120° and to insert Steiner points at appropriate positions (as in Figure 1b). Chang [5] gave a slightly more general variant: Select three vertices (terminals or Steiner points) of the current tree, insert a new Steiner point and remove the longest edge on any cycle created.

More recently, Beasley [4] proposed a similar approach based on enumerating connected subgraphs of the MST for $Z \cup S$ spanning four vertices. An SMT for the vertices of each subgraph is constructed and the corresponding Steiner points added to S in a greedy manner.

The application of geometric structures to heuristics for the Euclidean Steiner tree problem was initiated by Smith and Liebman [25]. They used a triangulation of

[3] Assuming that we are dealing with a minimization problem.

Z to aid the identification of Steiner points to insert. A similar but much more efficient heuristic for the Euclidean and rectilinear problems was given by Smith, Lee and Liebman [23, 24]. This first $O(n \log n)$ heuristic uses the well-known *Delaunay* triangulation (Figure 6a) to identify small sets of terminals with at most four terminals likely to be spanned by an FST in an optimal solution. The heuristic tree is constructed using a Kruskal-like greedy FST-concatenation algorithm.

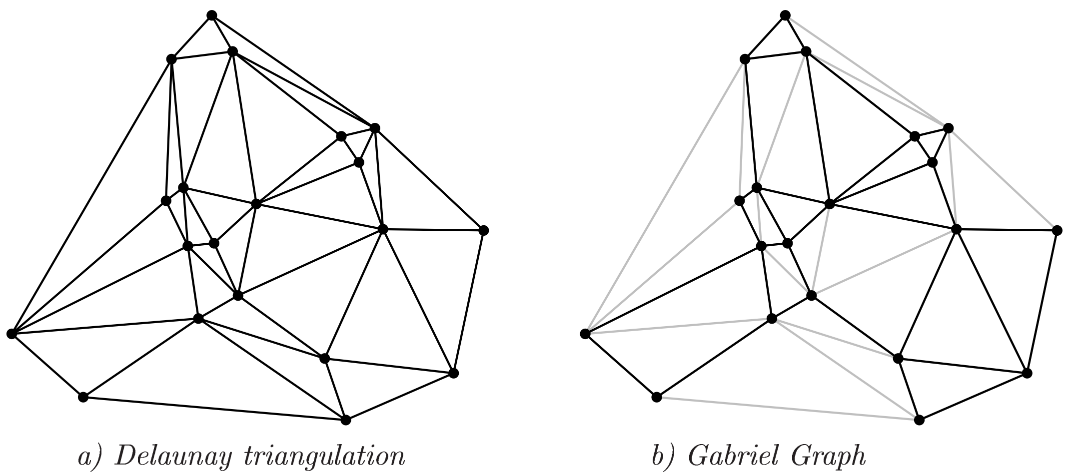


Figure 6: Proximity structures.

Zachariasen and Winter [37^B] generalized this approach in three directions. Firstly, alternative geometric structures for identifying "good" terminal subsets, namely Gabriel graphs (Figure 6b), relative neighbourhood graphs, and higher-order Voronoi diagrams were investigated. Secondly, terminal sets spanning up to K terminals (where K is a constant) were considered. Thirdly, the greedy FST-concatenation algorithm was significantly improved by adding an FST-insertion phase which iteratively improves the original tree constructed using the greedy FST-concatenation algorithm. All these improvements were added without breaking the $O(n \log n)$ running time barrier (only the running time constant increases). Extensive computational experiments showed that these new heuristics construct solutions that are better than those obtained by any other known $O(n \log n)$ heuristic, and, in fact, better than most other heuristics with much higher (or unknown) running time complexities.

4.2 Local Search Heuristics

Most heuristics for the Euclidean and rectilinear problems may (also) be characterized as local search heuristics. Local search is an heuristic search scheme which has been applied to a wide range of combinatorial optimization problems [1]. Starting from an initial solution (e.g., an MST), a local search algorithm generates a chain of solutions using a so-called *neighbourhood structure* which identifies a set of *neighbours* for every solution. The original form of local search, also denoted local optimization, used a greedy selection strategy in which only improving neighbours could be chosen.

To overcome the problem of being trapped in a local optimum, more sophisticated local search methods such as simulated annealing, tabu search and genetic algorithms were developed in the 1980s. However, applications of these methods to the Steiner tree problem have had limited success compared to other classical problems. This may be explained by the apparent lack of good (or natural) solution representations.

A survey of local search algorithms for the Euclidean Steiner tree problem is given in [34^C]. Heuristics are classified according to the underlying solution representations. The descriptions use local search terminology by presenting neighbourhood structures and neighbour selection strategies. Heuristics which represent solutions by their set of Steiner points seem to have the best performance among those suggested in the literature.

A novel FST based local search approach is presented in the same paper [34^C]. This approach uses the same two-phase strategy employed by the best exact algorithms (Section 3.2). However, FSTs are generated in an heuristic manner using a geometric structure, more precisely the Gabriel graph (Figure 6b). FST-concatenation is done by using local search on the list of generated FSTs. Three different neighbourhood structures are proposed and evaluated. Computational experience shows that this approach is very competitive from a cost-benefit point of view. The approach can easily be extended to the rectilinear problem; in this case heuristic FST generation may be replaced by fast "exact" FST generation [35^D].

An implementation of a tabu search algorithm for another geometric problem, the traveling salesman problem in the plane [36^F], supports the main conclusion of this section, namely that local search heuristics may benefit greatly from using geometric structures whenever applicable.

4.3 Arora's PTAS and Refinements

When Arora [2] in 1996 gave the first polynomial time approximation scheme (PTAS) for the Steiner tree problem in the plane, it followed (more or less) as a corollary on a similar result for the traveling salesman problem in the plane. The running time was simply $O(n^{C/\epsilon})$ where $\epsilon > 0$ is the relative performance guarantee and C is a constant. Later, Arora improved the original algorithm significantly [3] and very recently Rao and Smith presented an "optimal" algorithm in the sense that the asymptotic running time bound matches the $\Omega(n \log n)$ lower bound for the algebraic computation tree model [22]. However, the constants involved are still susceptible to improvement.

Currently, none of these algorithms have been implemented and whether they are competitive — from a practical point of view — to those described in this thesis is a very interesting open question. As a small appetiser we give a high-level description of their deterministic variants here. Let C denote a general constant which is not necessarily the same in the running time bounds mentioned below.

The main idea of all algorithms is to partition the plane recursively and to use dynamic programming to construct a solution with a guaranteed quality. The two most recent algorithms [3, 22] use a *quadtree* decomposition of the plane. A square surrounding the terminals is subdivided into four equal sized squares. Each of these is then subdivided recursively until each square contains at most one terminal. By rounding the coordinates of the terminals such that they lie on a grid with $O(n/\epsilon)$ lines, this decomposition can be guaranteed to have $O(\log(n/\epsilon))$ levels.

The quadtree decomposition can be *shifted* according to the horizontal and vertical positions of the terminals, giving a total of $O(n^2)$ possible shifts. Arora [3] showed that for at least half of these shifts there exists a tree spanning Z with the following three properties: i) The tree crosses each square in the decomposition at most $O(1/\epsilon)$ times, ii) the crossings only occur at set of $O((\log n)/\epsilon)$ equidistant points on the boundary of each square, also denoted "portals", and iii) the tree is a $(1+\epsilon)$-approximation to the SMT. Such a tree can be found in $O(n^3(\log n)^{C/\epsilon})$ time by using dynamic programming, i.e., by building the heuristic solution bottom-up in the quadtree.

In order to improve this running time bound to $O((1/\epsilon^C)\, n \log n)$, Rao and Smith [22] introduced *banyans* which generalize the notion of *spanners*. A $(1+\epsilon)$-spanner of Z is a subgraph of the complete Euclidean graph where for any pair of terminals z_i and z_j, the shortest path in the subgraph is at most $(1+\epsilon)$ times the Euclidean distance between z_i and z_j. Similarly, a $(1+\epsilon)$-banyan for Z is a graph whose vertices are a superset of Z and edges are line segments such that for any subset $Z' \subseteq Z$ a Steiner minimum tree in the banyan graph with Z'

as terminals is at most $(1+\epsilon)$ times the SMT for Z'. Rao and Smith showed that a $(1+\epsilon)$-banyan with $O(n)$ vertices (terminals and Steiner points) can be constructed in $O(n \log n)$ time.

By using Arora's quadtree decomposition approach to find an approximate SMT in the banyan graph, a $(1+\epsilon)$-approximation to the SMT for Z can be found in $O((1/\epsilon^C)\, n \log n)$ time. For fixed ϵ, this running time is $O(n \log n)$.

5 Conclusion

The development of algorithms for Steiner tree problems in the plane is yet another scholarly example of the importance of studying properties of optimal solutions when solving optimization problems. Both properties that are provably true in any optimal solution and properties *observed* to hold in solved problem instances are useful.

All algorithms in this thesis are heavily based on known properties of SMTs, in particular geometric properties. By using these properties exact and heuristic algorithms can eliminate the bulk of solutions investigated by trivial algorithms. The importance of using geometry when solving geometric problems is evident from the results presented in this thesis.

References

[1] E. H. L. Aarts and J. K. Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley & Sons, 1997.

[2] S. Arora. Polynomial Time Approximation Schemes for Euclidean TSP and other Geometric Problems. In *Proc. 37th Ann. Symp. on Foundations of Computer Science*, pages 2–13, 1996.

[3] S. Arora. Nearly Linear Time Approximation Schemes for Euclidean TSP and other Geometric Problems. In *Proc. 38th Ann. Symp. on Foundations of Computer Science*, pages 554–563, 1997.

[4] J. E. Beasley. A Heuristic for Euclidean and Rectilinear Steiner Problems. *European Journal of Operational Research*, 58:284–292, 1992.

[5] S. K. Chang. The Generation of Minimal Trees with a Steiner Topology. *J. Assoc. Comput. Mach.*, 19:699–711, 1972.

[6] S. E. Dreyfus and R. A. Wagner. The Steiner Problem in Graphs. *Networks*, 1:195–207, 1971.

[7] D. Z. Du and F. K. Hwang. A Proof of Gilbert and Pollak's Conjecture on the Steiner Ratio. *Algorithmica*, 7(2/3):121–135, 1992.

[8] M. R. Garey, R. L. Graham, and D. S. Johnson. The Complexity of Computing Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 32(4):835–859, 1977.

[9] M. R. Garey and D. S. Johnson. The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[10] E. N. Gilbert and H. O. Pollak. Steiner Minimal Trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

[11] S. L. Hakimi. Steiner's Problem in Graphs and Its Implications. *Networks*, 1:113–133, 1971.

[12] M. Hanan. On Steiner's Problem with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.

[13] D. S. Hochbaum, editor. *Approximation Algorithms*. PWS Publishing Company, Boston, 1996.

[14] F. K. Hwang. On Steiner Minimal Trees with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 30:104–114, 1976.

[15] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem.* Annals of Discrete Mathematics 53. Elsevier Science Publishers, Netherlands, 1992.

[16] F. K. Hwang and J. F. Weng. The Shortest Network under a Given Topology. *Journal of Algorithms*, 13:468–488, 1992.

[17] T. Koch and A. Martin. Solving Steiner Tree Problems in Graphs to Optimality. Technical Report SC-96-42, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, 1996.

[18] Z.-C. Liu and D.-Z. Du. On Steiner Minimal Trees with L_p Distance. *Algorithmica*, 7(2/3):179–191, 1992.

[19] A. Lucena and J. E. Beasley. A Branch and Cut Algorithm for the Steiner Problem in Graphs. *Networks*, 31:39–59, 1998.

[20] Z. A. Melzak. On the Problem of Steiner. *Canad. Math. Bull.*, 4(2):143–148, 1961.

[21] F.P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, second edition, 1988.

[22] S. B. Rao and W. D. Smith. Improved Approximation Schemes for Geometrical Graphs via "Spanners" and "Banyans". In *Proc. Thirtieth Annual ACM Symposium on Theory of Computing*, to appear.

[23] J. M. Smith, D. T. Lee, and J. S. Liebman. An $O(n \log n)$ Heuristic for the Rectilinear Steiner Minimal Tree Problem. *Engineering Optimization*, 4:179–192, 1980.

[24] J. M. Smith, D. T. Lee, and J. S. Liebman. An $O(n \log n)$ Heuristic for Steiner Minimal Tree Problems on the Euclidean Metric. *Networks*, 11:23–29, 1981.

[25] J. M. Smith and J. S. Liebman. Steiner Trees, Steiner Circuits and the Interference Problem in Building Design. *Engineering Optimization*, 4:15–36, 1979.

[26] W. D. Smith. How to Find Steiner Minimal Trees in Euclidean d-Space. *Algorithmica*, 7(2/3):137–177, 1992.

[27] E. A. Thompson. The Method of Minimum Evolution. *Annals of Human Genetics*, 36:333–340, 1973.

[28] D. Trietsch and F. K. Hwang. An Improved Algorithm for Steiner Trees. *SIAM Journal on Applied Mathematics*, 50(1):244–263, 1990.

[29] D. M. Warme. A New Exact Algorithm for Rectilinear Steiner Minimal Trees. Technical report, System Simulation Solutions, Inc., Alexandria, VA 22314, USA, 1997.

[30[E]] D. M. Warme, P. Winter, and M. Zachariasen. Exact Algorithms for Plane Steiner Tree Problems: A Computational Study. In D.-Z. Du, J. M. Smith, and J. H. Rubinstein, editors, *Advances in Steiner Trees*, Kluwer Academic Publishers, Boston, to appear.

[31] P. Winter. An Algorithm for the Steiner Problem in the Euclidean Plane. *Networks*, 15:323–345, 1985.

[32] P. Winter. Reductions for the Rectilinear Steiner Tree Problem. *Networks*, 26:187–198, 1995.

[33[A]] P. Winter and M. Zachariasen. Euclidean Steiner Minimum Trees: An Improved Exact Algorithm. *Networks*, 30:149–166, 1997.

[34[C]] M. Zachariasen. Local Search for the Steiner Tree Problem in the Euclidean Plane. *European Journal of Operational Research*, to appear.

[35[D]] M. Zachariasen. Rectilinear Full Steiner Tree Generation. *Networks*, to appear.

[36[F]] M. Zachariasen and M. Dam. Tabu Search on the Geometric Traveling Salesman Problem. In I. H. Osman and J. P. Kelly, editors, *Metaheuristics: theory and applications (Proceedings from Metaheuristics International Conference, Colorado)*, pages 571–587, 1995.

[37[B]] M. Zachariasen and P. Winter. Concatenation-Based Greedy Heuristics for the Euclidean Steiner Tree Problem. *Algorithmica*, to appear.

Summary (in Danish)

Topologisk netværksdesign består i at planlægge layoutet af et netværk under betingelser vedrørende dets topologi. Applikationerne omfatter transport- og kommunikationsnetværk, hvor konstruktionsomkostningerne typisk er forbundet med netværkets knuder og/eller kanter.

Steiner problemet er et af de fundamentale problemer indenfor topologisk netværksdesign. Problemet består i at sammenbinde (en delmængde) af knuderne, således at der er en vej imellem alle knudepar; den samlede omkostning af de valgte kanter skal minimeres. Det originale Steiner problem blev stillet som et rent geometrisk problem. For en given mængde punkter (terminaler) i planen skal et træ, der sammenbinder terminalerne, konstrueres, således at den totale længde af alle liniestykkerne i træet minimeres. Det *euklidiske* Steiner træ problem benytter den almindelige euklidiske (eller L_2) afstand imellem liniestykkernes endepunkter, mens det *rektlineære* problem benytter den rektlineære (eller L_1) afstand.

Denne afhandling omhandler algoritmer til løsning af euklidiske og rektlineære Steiner træ problemer. Specielt er en række nye eksakte og heuristiske algoritmer evalueret ved udførsel af omfattende eksperimenter. Afhandlingen dækker det fulde spektrum af algoritmer, især for det euklidiske problem: Hurtige grådige algoritmer med $O(n \log n)$ værste-tilfælde kørselstider, gode lokalsøgningsalgoritmer som konstruerer løsninger af høj kvalitet og effektive eksakte algoritmer, som kan finde optimale løsninger til probleminstanser med mere end 2000 terminaler. De nye heuristikker finder bedre løsninger hurtigere end alle andre kendte heuristikker og den nye eksakte algoritme kan løse probleminstanser, der er mere end 10 gange større end hidtil løste instanser.

Algoritmernes effektivetet skyldes fortrinsvis en strukturel egenskab ved de optimale løsninger for disse to problemer. Det optimale Steiner træ kan splittes op i såkaldte fulde Steiner træer (FSTer) hvori alle terminaler er blade. Disse FSTer er typisk meget små (udspænder sjældent mere end seks terminaler) og har mange velkendte egenskaber, der kan udnyttes effektivt ved anvendelse af geometriske strukturer.

Afhandlingen er en samling af seks artikler. Fem af disse omhandler det euklidiske og rektlineære Steiner træ problem, mens den sidste præsenterer en tabusøgningsalgoritme for et andet geometrisk problem, den omrejsende sælgers problem i planen. Afhandlingen indledes med en kort introduktion til algoritmer for Steiner problemet i planen.

Research Papers

A P. Winter and M. Zachariasen. Euclidean Steiner Minimum Trees: An Improved Exact Algorithm. *Networks*, 30:149–166, 1997.

B M. Zachariasen and P. Winter. Concatenation-Based Greedy Heuristics for the Euclidean Steiner Tree Problem. *Algorithmica*, to appear.

C M. Zachariasen. Local Search for the Steiner Tree Problem in the Euclidean Plane. *European Journal of Operational Research*, to appear.

D M. Zachariasen. Rectilinear Full Steiner Tree Generation. *Networks*, to appear.

E D. M. Warme, P. Winter, and M. Zachariasen. Exact Algorithms for Plane Steiner Tree Problems: A Computational Study. In D.-Z. Du, J. M. Smith, and J. H. Rubinstein, editors, *Advances in Steiner Trees*, Kluwer Academic Publishers, Boston, to appear.

F M. Zachariasen and M. Dam. Tabu Search on the Geometric Traveling Salesman Problem. In I. H. Osman and J. P. Kelly, editors, *Metaheuristics: theory and applications (Proceedings from Metaheuristics International Conference, Colorado)*, pages 571–587, 1995.